PC 128 - PC 128S - PC 1

PRODEST

LA PRIMA E UNICA
RIVISTA INDIPENDENTE
PER GLI UTENTI DEI
[illegible] OLIVETTI PRODEST

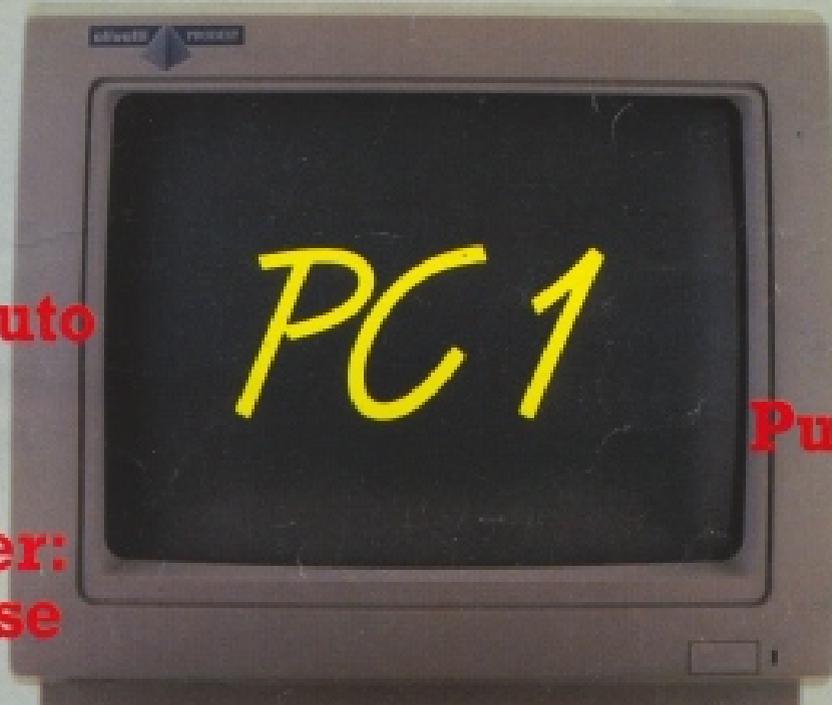
GRUPPO EDITORIALE JACKSON
DIVISIONE PERIODICI

Numero 6 - Ottobre-Novembre '87

Spedizione in abbonamento postale Gruppo III/70

PC 1

Benvenuto PC1

File Manager: data base

Desk Top Publishing con il PC 128S

NOVITÀ ASSOLUTA
Il nuovo PC 1
MS-DOS compatibile

PROVA SU STRADA DEL PC 1

Verifica delle caratteristiche tecniche del nuovo computer prodotto interamente in Italia

Con il lancio sul mercato di PC1 l'Olivetti Prodest è ora in grado di offrire al grande pubblico degli utenti la più completa gamma di prodotti Home Computer disponibile sul mercato, che copre tutte le esigenze, dal primo passo di avvicinamento all'informatica sino agli utilizzi per piccole attività professionali, commerciali o artigianali.

Scheda tecnica del PC1

Il computer PC1 è un computer decisamente molto potente, in quanto già «di serie» possiede ben 512 Kbyte di memoria RAM (Random Access Memory, memoria ad accesso casuale), espandibile fino a 640 Kbyte, per mezzo di un modulo di espansione.

Per quanto riguarda la scelta del sistema operativo, MS/DOS 3.2® *(® MS-DOS è un Trademark della Microsoft Inc.)* esso è attualmente il più diffuso in campo professionale.

Sono in commercio, compatibili con questo sistema operativo, moltissimi programmi.

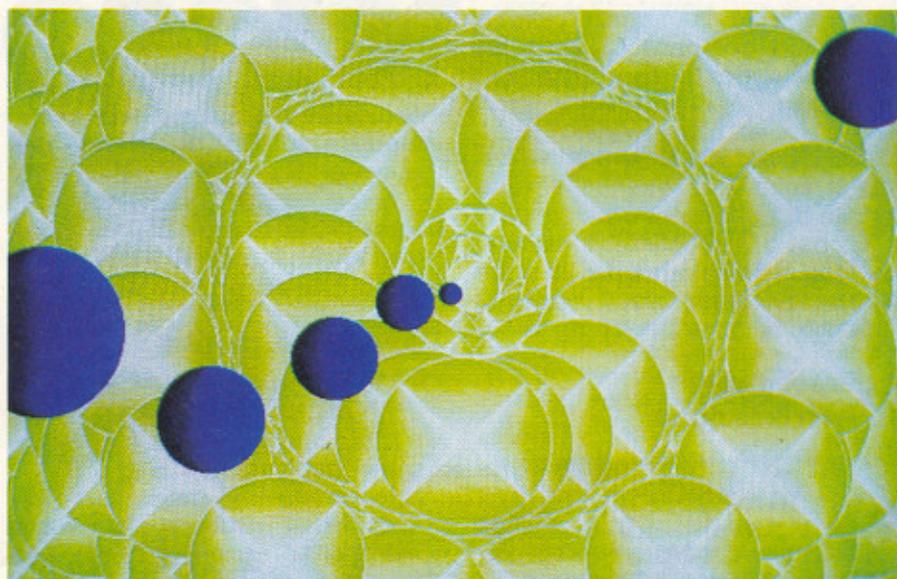
L'Olivetti Prodest, operazione che raramente avviene nel campo dell'informatica, ha fatto uscire, insieme al PC1, più di 100 nuovi programmi creati per rendere più semplice e completo l'uso di questo nuovo computer.

Naturalmente, questi programmi, attenendosi allo standard MS-DOS, saranno utilizzabili da chiunque possieda un computer ad esso compatibile.

Il microprocessore

Il microprocessore, vero cuore pulsante di ogni computer, è il V40 della NEC.

Questo microprocessore è completamente compatibile con il 8088 dell'Intel (nota industria di componenti elettronici), ma ha in più una particolarità: pur lavorando alla stessa velocità è in grado di eseguire una stessa istruzione in minor tempo.

Questa apparente contraddizione è resa possibile dalla scelta fatta dalla NEC, spiegata in soldoni, di evitare una codifica «software» delle istruzioni che il microprocessore riceve, sostituendo una codifica di tipo «hardware», decisamente più veloce.

Anche il microprocessore, quindi, è quanto di meglio il mercato mondiale mette a disposizione per la costruzione di computer.

Altra caratteristica del V40 è la possibilità di settare, cioè modificare, la velocità via software: non è quindi necessario aprire il computer per modificare il clock da 4.77 Megahertz a 8 Megahertz o viceversa.

È chiaro che alla frequenza di 8 Megahertz i programmi, di gestione grafica o che eseguono calcoli matematici, verranno eseguiti più velocemente.

La tastiera

La tastiera del PC1 ha un tocco molto particolare, e aiuta la digitazione molto veloce.

Questa caratteristica è particolarmente utile nel caso di uso del computer per la programmazione o per la gestione di testi.

La tastiera del PC1, di tipo QWERTY, conta di ben 83 tasti. Essa è divisa nella sezione alfanumerica, dove trovano posto tutte le lettere e i caratteri di interpunzione, il tastierino numerico, molto utile quando il computer viene usato con un qualsiasi programma di gestione amministrativa, e una parte, rintracciabile nella zona superiore della tastiera stessa, che prevede 10 tasti funzione.

Nella parte in alto a destra, accanto ai tasti funzione poco sopra accennati, sono presenti tre spie indicanti lo stato della macchina (on spia accesa, off spia spenta), il modo maiuscolo (spia accesa lettere visualizzate in maiuscolo, spia spenta lettere visualizzate in minuscolo) e, per finire, la terza spia indica se il tastierino numerico può essere usato come tale o i tasti possono essere usati nella loro seconda funzione, quella di tasti cursore.

In quest'ultimo modo ad ogni tasto verrà associato uno spostamento del cursore nel video.

Ad esempio il tasto corrispondente al numero 4 porterà il cursore alla sinistra di una posizione, il 6 alla destra di una posizione, il tasto 8 in su di una riga e il tasto 2 in basso, sempre di una riga.

La grafica

Lo standard grafico adottato dal nuovo prodotto del Olivetti Prodest è il CGA.

Questo standard permette di visualizzare, con due colori, dei grafici con una risoluzione di 640×200 punti, mentre se vogliamo usare quattro colori, la massima risoluzione possibile sarà 320×200.

In sedici colori la risoluzione di-

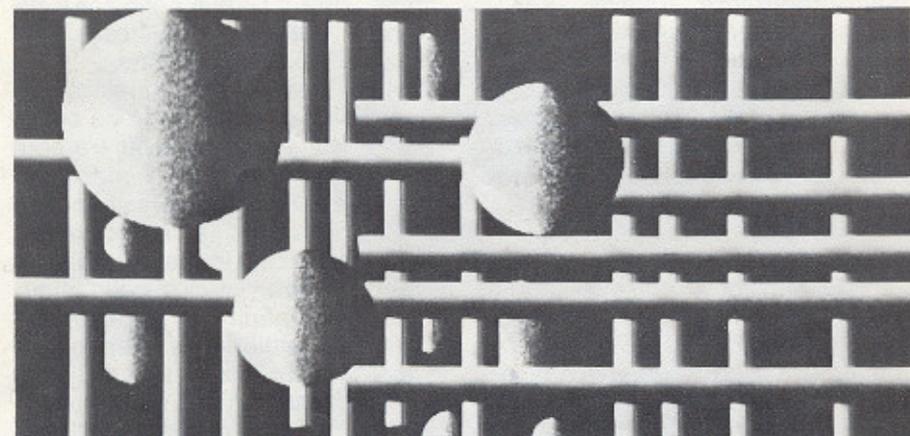
minuirà (160×200), ma l'effetto cromatico compenserà la diminuita accuratezza del grafico.

Il testo

Nel modo testo i caratteri visualizzati sono stati ottenuti da una griglia di punti di otto per otto.

È possibile scegliere la visualizzazione di 40 colonne per 25 righe oppure è disponibile il modo 80 colonne, sempre per 25 righe.

I dispositivi di memorizzazione

Il dispositivo usato per memorizzare sia i programmi che i dati elaborati è un dischetto di plastica rigida, dalle dimensioni di 3.5 pollici, pari a circa 9.2 centimetri.

Il tipo usato è quello che tecnicamente si chiama Doppia Faccia/Doppia Densità ed ha una capacità di ben 720 Kbyte.

Il disk driver è di tipo slim cioè ribassato.

Anche la scelta del tipo di floppy disk driver è stata improntata dalla ricerca di ciò che di più avanzato la tecnologia mette a disposizione nel campo degli elaboratori. Questa misura di disco (3.5 pollici) non è una scelta casuale, ma è il nuovo standard nel campo dei sistemi compatibili MS-DOS.

Inoltre, è anche disponibile un driver da 5.25 pollici. Inoltre sarà disponibile un disco fisso (hard disk nella dizione inglese), capace di memorizzare 20 Megabyte.

Il sistema Operativo

Dopo aver parlato tanto di sistema operativo, diamo qualche notizia in più sul sistema implementato sul PC1. Il sistema operativo è l'MS-DOS in versione 3.2

Questa è l'ultima versione in commercio di tale sistema operativo: anche in questo il PC1 dell'Olivetti Prodest è all'avanguardia.

Joystick e mouse

Questi due utilissimi accessori sono disponibili anche per il PC1, e il loro funzionamento è in emulazione dei cursori.

Il joystick è collegato in modo che il computer risponda indifferentemente alla pressione dei tasti cursore o al movimento del joystick stesso. Il mouse Olivetti Prodest è invece fornito di un programma "DRIVER" che permette la piena funzionalità della periferica che ha le medesime caratteristiche del mouse IBM. L'unica diversità è che il mouse Olivetti Prodest usa la porta joystick per il collegamento alla CPU.

Interfacce hardware disponibili

Vi sono varie interfacce disponibili per comunicare con dispositivi esterni quali: plotter, stampanti ecc.

Tra le varie disponibili ricordiamo:

- Interfaccia stampante parallela Centronics.
- Interfaccia seriale standard RS 232.
- Uscita sonora HI-FI.
- Floppy disk driver da 5.25 pollici: il vecchio standard attualmente sostituito dallo standard a 3.5 pollici.
- Expansion bus (compatibile IBM).

Opzioni disponibili

Tramite l'uso della Expansion bus è possibile connettere un box, il quale potrà contenere due schede small/half size: di misura ridotta cioè.

Le opzioni previste sono le seguenti:

- Scheda di memoria aggiuntiva, per portare la memoria sino al massimo consentito, 640 Kbyte.
- Scheda per la rete locale di elaboratori.
- Scheda per il modem.
- Scheda grafica avanzata EGA.
- Controllore per l'uso di un Compact Disk come banca dati consultabile con il computer. Naturalmente il disco conterrà non informazioni atte alla riproduzione sonora di un brano di musica classica o rock, ma informazioni, per esempio, sulle sentenze riguardanti un certo argomento emanate in Italia.
- Adattatore televisivo e telematico.
- Controllore dell'unità a disco fisso (spiegata poco sopra a riguardo dei dispositivi di memorizzazione).
- Adattatore per Pagine Gialle Elettroniche (PGE).

Conclusione

Questa prova, confortata soprattutto dalle eccellenti caratteristiche tecniche, non poteva che farci spezzare una lancia a favore di questo nuovo prodotto dell'Olivetti Prodest, che si inserirà in un mercato, chissà perché finora ignorato dagli altri grandi produttori di computer: il mercato delle piccole attività, dei piccoli studi professionali e delle piccole e medie attività commerciali o artigianali.

I DUE BASIC DEL PC 128

Continua il nostro viaggio attraverso i comandi Basic implementati sul PC 128

6ª Parte

Per chi non avesse avuto modo di leggere gli articoli precedenti, ricordiamo che la struttura esplicativa adottata è identica per ogni comando e segue questa forma:

NOME DEL COMANDO
TIPO: Ovvero l'ambito d'uso del comando.
SINTASSI: ovvero la forma corretta del comando, composta dalla parola chiave e dai suoi parametri.
COMMENTO: Ovvero la spiegazione del comando e dei suoi usi.

DEFUSR

TIPO: Comando

SINTASSI: DEFUSRn.=indirizzo

COMMENTO:

Dove n. è il numero di chiamata di un programma in linguaggio macchina e indirizzo è l'indirizzo di partenza dello stesso.

Per mezzo di questa istruzione si possono definire fino a 10 funzioni utente in linguaggio macchina, numerate da 0 a 9.

Lo spazio necessario in memoria, può essere ricavato per mezzo dell'istruzione CLEAR.

Ogni routine in linguaggio macchina deve terminare con l'istruzione RTS (Ritorno da una Soubroutine), e non deve alterare i registri S e DP.

In questa sede non è possibile approfondire l'uso di questo comando, il quale è del tutto inutile se non si è in possesso di una buona tecnica di programmazione in linguaggio macchina, pertanto sconsigliamo il suo utilizzo da parte di programmatori meno che esperti.

DEF USR4=&H3FFF Definisce la funzione numero 4 all'indirizzo &H3FFF.

DELETE

TIPO: Comando o Istruzione.

SINTASSI: DELETE n. R1-n.R2,n.

COMMENTO:

Dove n.R1 è la riga di partenza; n.R2 è la riga d'arrivo del blocco da cancellare; n. è il numero di riga da cui ricominciare il programma.

Tramite questo comando, utilizzabile pure come istruzione, è possibile cancellare blocchi definiti di un programma.

L'estensione di questi blocchi va dalla riga singola all'intero programma.

Se utilizzato come istruzione, DELETE cancella il blocco di righe in argomento e fa ricominciare il programma, dal numero di riga riportato nel terzo parametro, n. del comando.

L'uso dei parametri nel comando DELETE è molto elastico, e grazie a questa versatilità il co-

mando stesso può essere usato in diversi modi, a secondo delle proprie esigenze; di seguito vediamo alcuni esempi:

DELETEn.R1
Cancella la riga n. R1
DELETE.
Cancella la riga corrente.
DELETEn.R1-n.R2
Cancella il blocco di righe delimitato da n.R1-n.R2, comprese le righe di riferimento.
DELETEn.R1-
Cancella tutte le righe da n.R1 alla fine.
DELETE-n.R2
Cancella tutta la parte di programma compresa tra la prima riga e n.R2.
DELETE-
Cancella tutto il programma.
DELETEn.R1-n.R2,n.
Cancella il blocco di righe delimitato da n.R1-n.R2, e poi ricomincia ad eseguire il programma a partire dalla riga n.

DENSITY

TIPO: Istruzione.

SINTASSI: DENSITY n.drive, densità

COMMENTO:
Dove n.drive è il numero del dirve a cui ci si riferisce (può essere compreso tra 0 e 4); densità deve essere un numero uguale a 1 per la singola densità, oppure a 2, per la doppia densità.

Per mezzo di questo comando, si può fissare la densità di registrazione dei dati del drive in argomento.

C'è da notare, che all'attivazione, un drive assume la densità del dischetto in esso contenuto.

DENSITY1,1 Imposta il drive in densità singola.

DEVICE

TIPO: Istruzione.

SINTASSI: DEVICE «nome periferica»

COMMENTO:
Il comando DEVICE è utilizzato per cambiare la periferica di default: questo viene solitamente impostato automaticamente dal computer alla sua accensione. Per esempio nel Basic 128 la periferica di default è il drive0, mentre nel Basic 1 è il registratore.

È immediato che non sempre queste sono le periferiche a noi più utili, e pertanto è facile ovviare alla situazione per mezzo del comando DEVICE.

DEVICE«1:» Indica nel drive 1 il nuovo dispositivo di default.
DEVICE«CASS:»

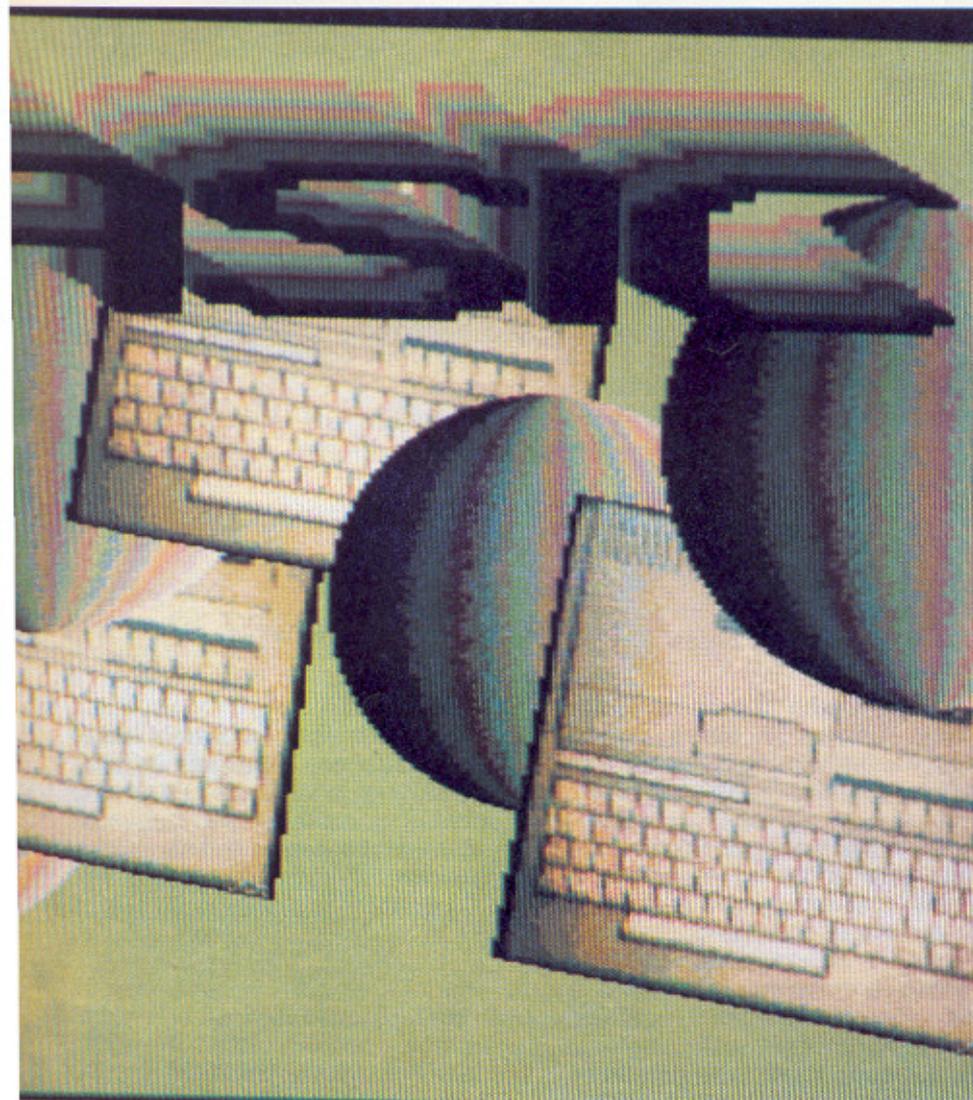
DIM

TIPO: Istruzione.

SINTASSI: DIM var(ind), var(ind)

COMMENTO:
Dove var è il nome di una variabile scritta nel formato consentito dal Basic, e ind è il valore massimo raggiungibile dall'indice.

Il comando DIM viene utilizzato per dimensionare le matrici, ciò vuol dire che si riserva loro un ulteriore spazio in memoria. Infatti si devono dimensionare solo le variabili con indice superiore a 10.

Durante l'esecuzione del comando DIM, gli elementi delle matrici di tipo numerico vengono inizializzate a 0 mentre gli elementi delle matrici stringa, vengono inizializzate al valore «», ossia di stringa nulla.

È importante notare che il dimensionamento di una matrice deve esser fatto solo una volta, altrimenti apparirà il messaggio d'errore «Duplicate Definition in n.riga».

Se una matrice viene invece usata in un campo esterno a quello di dimensionamento, si avrà il messaggio d'errore «Bad Subscrit».

Vediamo un semplice programma, che dimostra l'uso del comando DIM e delle matrici, o array, sia monodimensionali, simili cioè a una lista, che bidimensionali, simili cioè a una tabella.

```
10    CLS
20    DIMVAR1$(10,4)VAR2$(28)
30    FOR A=0 TO 4
40    FOR B=0 TO 10
50    READ VAR1$(B,A)
60    NEXTB,A
70    FOR C=0 TO 4
80    FOR D=0 TO 10
90    S=S+1
100   PRINT VAR1$(D,C);
110   IF S=0 OR Q>28 THEN 130
120   IF   S=INT(S/2)*2   THEN
      Q=Q+1:
      VAR2$(Q)=VAR1$(D,C)
130   NEXTD,C
140   PRINT:PRINT
150   FOR G=1 TO 28
160   PRINT VAR2$(G);
170   NEXT G
180   DATA A,B,C,D,E,F,G,H,I,L,M,
      N,O,P,Q,R,S,T,U,V,Z
190   DATA A1,B1,C1,D1,E1,F1,
      G1,H1,I1,L1,M1,N1,O1,P1,
      Q1,R1,S1,T1,U1,V1,Z1
200   DATA A2,B2,C2,D2,E2,F2,
      G2,H2,I2,L2,M2,N2,O2
```

Come si può vedere è semplice costruire una tabella bidimensionale in cui caricare dei dati, ciò viene fatto nel ciclo dalla riga 30 alla riga 60, per poi poterli utilizzare nei modi a noi più utili: vedi le righe di programma dal numero 70 al numero 130.

Nella riga 120, c'è un semplice sistema per selezionare i numeri pari. Questo è utilizzato per caricare una matrice monodimensionale, VAR2$() con solo alcuni de valori della variabile bidimensionale VAR1$(): più esattamente, i valori considerati sono quelli relativi al contenuto della variabile VAR1$() nel momento in cui il valore della variabile S è un numero pari.

DIR

TIPO: Comando.

SINTASSI: DIR«n.drive: *nomefile* estensione»

COMMENTO:

Dove n.drive è il numero dato al drive di cui vogliamo vedere la directory; *nomefile* è il nome del file di cui si vogliono le informazioni, ed estensione indica il tipo di file indicato.

DIR è il comando da utilizzare nel caso in cui si necessiti di informazioni sui file residenti in un dischetto.

Se n.drive viene omesso, il drive considerato sarà quello di default, lo 0. Ciò è logicamente valido solo nel caso in cui non ci siano state delle variazioni causate dall'uso del comando DEVICE, in tal caso sarà il numero in argomento a questo comando a determinare il drive di default.

Nel caso in cui non sia presente il *nomefile*, il comando visualizzerà tutti i file aventi l'estensione indicata, se invece è l'estensione a essere omessa, i file mostrati saranno tutti quelli aventi lo stesso *nomefile*.

Se il *nomefile* è incompleto, verranno visualizzati tutti i file aventi per radice il *nomefile* stesso. Nel caso in cui siano assenti tutti i parametri del comando, verranno visualizzati tutti i file presenti sul dischetto.

L'header che appare nella directory, riporta: il nome del dischetto, il suo numero, la sua densità e lo spazio libero espresso in Kbyte.

Ogni file viene visualizzato nel seguente formato: *nomefile*, estensione, tipo di file, tipo di dati, dimensione del file espresso in Kbyte e, se esistente, un commento di massimo otto byte di estensione. Questi dati vengono visualizzati tramite delle sigle, le quali hanno rispettivamente le seguenti corrispondenze:

nomefile (massimo 8 caratteri)

ESTENSIONE

BAS-prg. Basic
DAT-dati
BIN -binario
MAP-immagine
TRA-disegno

TIPO FILE

B- prg. Basic
D- dati
A- prg. in Assembler
M- prg. in L.M.

TIPO DI DATI

A- ASCII
B- binario

DIR«0:PROVA1.BAS»
visualizza sullo schermo i dati relativi al file denominato PROVA1.BAS dal dischetto inserito nel drive 0.

DIR«0:MAP»
visualizza tutti i file contenuti nel drive 0, aventi MAP come estensione.

DIR«S»
visualizza tutti i file inizianti per S, contenuti nel dischetto posto nel drive corrente.

DIR
visualizza tutti i file contenuti nel drive corrente.

DIRP

TIPO: Comando

SINTASSI:
DIRP«n.drive:*nomefile*.
estensione»

COMMENTO:

Tramite questo comando, dalla sintassi identica a DIR, è possibile direzionare la directory di un dischetto, invece che sullo schermo del monitor, verso la stampante parallela.

Oltre alla sintassi, anche il significato delle sigle, è identico a quelle del comando DIR.

DIRP«1.PROVA1.BAS»
dirige la directory del file PROVA1.-BAS, del drive 1, alla stampante.

DO...LOOP

TIPO: Istruzione di ciclo d'iterazione.

SINTASSI: DO... (istruzioni) ... LOOP.

COMMENTO:

Dove DO è l'inizio del ciclo, (istruzioni) il corpo del ciclo e LOOP il rinvio all'inizio del ciclo.

Per mezzo del ciclo DO...LOOP, si può creare una specie di trappola, ovvero una parte di programma funzionante in modo continuato e ripetuto, fino a quando non si verifica una situazione voluta dal programmatore. Solo in questo caso, grazie al comando EXIT sarà possibile uscire dal ciclo.

Il comando EXIT, fa riprendere l'esecuzione del programma dalla prima riga successiva al comando LOOP.

I cicli possono essere nidificati, avendo però cura di chiudere sempre per primi i cicli interni: un po' quello che si deve fare nei cicli FOR... NEXT.

Oltre al comando EXIT, anche i comandi GOTO, ON...GOTO e THEN possono permettere l'uscita da un ciclo, ma è opportuno usarli solo in casi eccezionali, altrimenti si viene a perdere la necessaria chiarezza dei programmi ben strutturati.

Si deve inoltre precisare che non è possibile entrare in un ciclo DO...LOOP in un qualsiasi suo punto, in tale caso infatti, apparirà il messaggio d'errore «Loop Without DO». Ciò accade perché il ciclo può avere inizio solo dal comando DO, e pertanto se al momento dell'istruzione LOOP l'interprete Basic scopre che non è stato prima letto il comando DO, ne segnala immediatamente l'errore.

Una delle applicazioni più utili dei cicli DO...LOOP, è il controllo degli input:

```
10   CLS
20   DO
30   CLS
40   INPUT«IMMETTERE UN NU-
     MERO (0-9)»;A$
50   IF A$=«8» THEN EXIT
60   CLS
70   PRINT«MI DISPIACE MA HAI
     SBAGLIATO!»
80   FOR A=0 TO 15000: NEXT A
     LOOP
100  CLS
110  PRINT«BRAVO HAI VINTO!!!»
120  END
```

Notare l'uso della variabile stringa A$ in luogo di una qualsiasi variabile numerica. Per comprenderne appieno il motivo provate a trasformare A$ in A, logicamente togliendo anche le virgolette dalle assegnazioni, e provate a rispondere all'input per mezzo di una lettera invece di un numero.

Il ciclo FOR...NEXT in riga 80, è un semplice ciclo d'attesa.

DOS

TIPO: Comando.

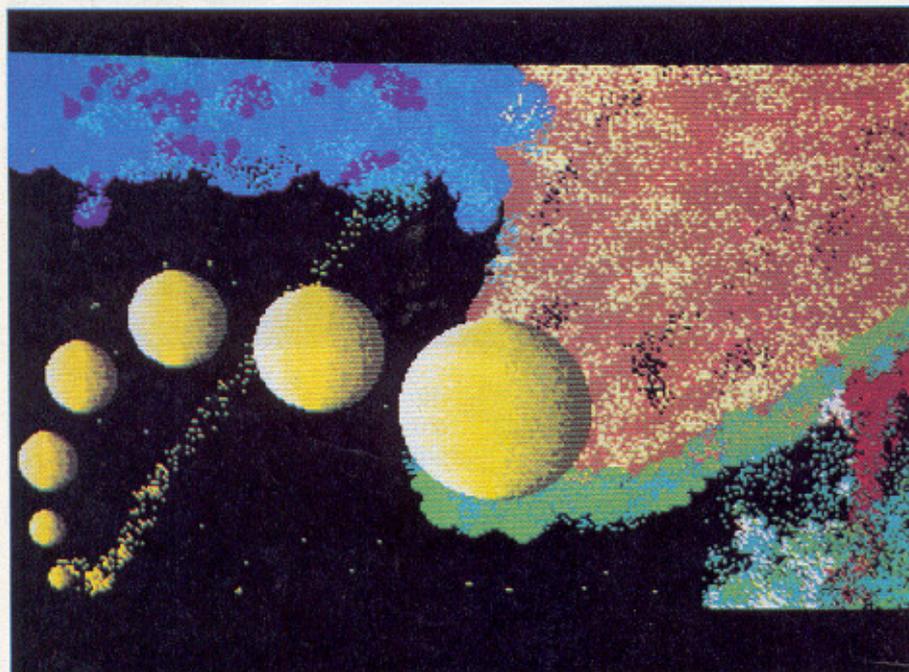
SINTASSI: DOS.

COMMENTO:
Ritorna alla pagina iniziale, perdendo il contenuto della memoria corrente, per poter caricare il DOS del Basic 1.

DRAW

TIPO: Istruzione.

SINTASSI: DRAW stringa di caratteri.

COMMENTO:
Dove la stringa di caratteri è un insieme di comandi elementari, con relativo argomento, che definisce il modo in cui sarà eseguito il disegno quando il Basic attiverà il comando DRAW.

Di seguito viene riportato un elenco dei comandi elementari, con i relativi significati:

Un Spostamento verso l'alto.
Dn Spostamento verso il basso.
Rn Spostamento verso destra.
Ln Spostamento verso sinistra.
En Spostamento diagonale verso l'alto a destra.
Fn Spostamento diagonale verso il basso a destra.
Gn Spostamento diagonale verso il basso a sinistra.
Hn Spostamento diagonale verso l'alto a sinistra.
Mx,y Spostamento assoluto o relativo, dove x è la colonna e y la riga. Se lo spostamento è di tipo relativo, è obbligatorio far precedere x e y dal segno – o +.
Sn Fattore di scala, n può essere compreso tra 1 e 63. Il valore del parametro n, moltiplicato per il valore degli altri comandi elementari, dà il reale valore di spostamento. Affinché il fattore di scala sia pari a 1:1, il parametro deve essere pari a 4.

An Imposta l'angolo di rotazione per mezzo del valore di n, il quale può essere compreso tra 0 e 3. Il valore del parametro n corrisponde alle seguenti impostazioni di rotazioni relative: 0 verso l'alto; 1 verso destra; 2 verso il basso, 3 verso sinistra.
Cn Imposta il colore del tracciato. Il parametro n, può variare da –16 a 15.
B Posto davanti ad un altro comando, esegue uno spostamento, senza però visualizzarlo per mezzo di un segno.
N Posto davanti ad un comando, permette di ritornare alla posizione di partenza, dopo aver però eseguito l'istruzione che lo segue.
Xn$ Permette di eseguire tracciati definiti da stringhe del tipo n$. I valori di n possono essere introdotti anche per mezzo di variabili ricorrendo ad una assegnazione del tipo: G=x Dove x è una variabile numerica precedentemente definita.

All'interno della stringa di istruzioni elementari, quest'ultimi possono essere divisi o da uno spazio o da un punto e virgola. È possibile lasciare uno spazio anche fra i comandi e il loro argomento.

DRAW«C8;B M80,58;L35;U40;
H20»

DSKF

TIPO: Funzione.

SINTASSI: DSKF(n.drive)

COMMENTO:
Dove n. drive è il numero del drive di cui si vuole l'informazione.

Per mezzo di questo comando è possibile ottenere il numero esatto di Kbyte liberi sul dischetto inserito nel drive in argomento. Il numero del drive è obbligatorio.

```
10 CLS
20 A=DSKF(0)
30 PRINT A
40 END
```

DSKI$

TIPO: Funzione.

SINTASSI:
DSKI$(n.drive,traccia,settore).

COMMENTO:
Dove n. drive è il numero del drive in cui è inserito il dischetto da leggere; traccia è un numero, o una variabile numerica intera, compresa tra 0 e 79; settore è un numero, o una variabile numerica intera, compreso tra 1 e 16.

Per mezzo del comando DSKI$, è possibile leggere il contenuto di

una parte di un dischetto, delimitata appunto dai valori di traccia e settore. Il risultato della lettura, sarà una stringa di caratteri composta di 128 byte, se il dischetto è in densità singola, oppure di 255 byte se il dischetto è in doppia densità.

L'utilità del comando DSKI$, è da mettere in relazione al comando DSK0$, per mezzo del quale è possibile cambiare i dati precedentemente letti, appunto per mezzo del comando DSKI$.

Come si può notare, questi due comandi sono molto simili, concettualmente non nella sintassi, ai comandi di manipolazione della memoria PEEK e POKE. Per mezzo dei quali è possibile, rispettivamente, leggere e scrivere all'interno della memoria del computer.

Di seguito presentiamo un piccolo programma che vi metterà in grado di leggere i contenuti più segreti dei vostri dischetti, così da permettervi d'impratichirvi nella lettura dei dati in essi contenuti.

```
10  CLS
20  FOR A=0 TO 79
30  FOR B=1 TO 16
40  PRINT«Per uscire premere
    'Q'»
50  PRINT
60  PRINT«TRACCIA n. »;A
70  PRINT«SETTORE n.  »;B
80  PRINT
90  PRINT DSKI$(0,A,B)
100 FOR X=1 TO 1000: NEXT X
110 CLS
120 ON KEY=«Q» GOTO 140
130 NEXT B,A
140 END
```

La linea interessata alla stampa delle informazioni contenute sul dischetto è la 90. La linea 120 è un utile esempio di come sia facilmente controllabile l'input al programma. Infatti, per mezzo di questa riga, è possibile uscire dai cicli in qualsiasi momento, senza che per questo si debba resettare il computer.

DSKINI

TIPO: Comando.

SINTASSI:

DSKINIn.drive,intreccio,«nome»

COMMENTO:

Dove n.drive è il numero del drive contenente il dischetto su cui si vuole operare; intreccio è un numero, per quanto riguarda la nostra trattazione però questo è di nessuna utilità e pertanto può essere omesso. Il terzo parametro è il nome da dare al dischetto, e può essere una stringa di, al massimo, otto caratteri.

Il comando in questione si utilizza per formattare i dischetti da utilizzare sul PC 128.

Se il dischetto è difettoso, il computer risponderà con un messaggio d'errore del tipo «Input/Output Error».

Dei tre parametri del comando DSKINI solo il primo, n.drive, è obbligatorio, gli altri due potranno essere omessi. Se prima del lancio del comando, verrà impartito al computer l'ordine di verificare la formattazione per mezzo dello statement VERIFY ON, l'operazione verrà svolta, ma impiegherà un tempo maggiore per essere portata a fine.

DSKINI0,«WORK»
Formatta il dischetto contenuto nel drive 0, con il nome di WORK.

DSKINI0
Formatta il dischetto inserito nel drive 0.

DSKO$

TIPO: Istruzione.

SINTASSI:
DSKO$n.drive, traccia, settore, stringa.

COMMENTO:

Dove n.drive è il numero del drive in cui è contenuto il dischetto su cui si vuole operare; traccia è un numero o una variabile intera, compreso tra 0 e 79, indicante la traccia interessata; settore è un numero, compreso tra 1 e 16, o una variabile intera indicante il settore interessato e stringa è una stringa di massimi 128 caratteri, se il dischetto è a densità singola, oppure di 255 caratteri se il dischetto è a densità doppia. Se la dimensione della stringa è inferiore ai 128 o ai 255 caratteri, la parte rimanente sarà riempita con caratteri di codice ASCII 0.

Il comando consente la scrittura di dati all'interno di uno spazio delimitato da un settore e da una traccia, in un determinato dischetto.